一、数据库备份分类

1、逻辑与物理角度

从物理与逻辑的角度，备份可分为物理备份、逻辑备份。

物理备份：对数据库操作系统的物理文件（如数据文件、日志文件）的备份

冷备份 (脱机备份) :是在关闭数据库的时候进行的

热备份 (联机备份) :数据库处于运行状态，依赖于数据库的日志文件

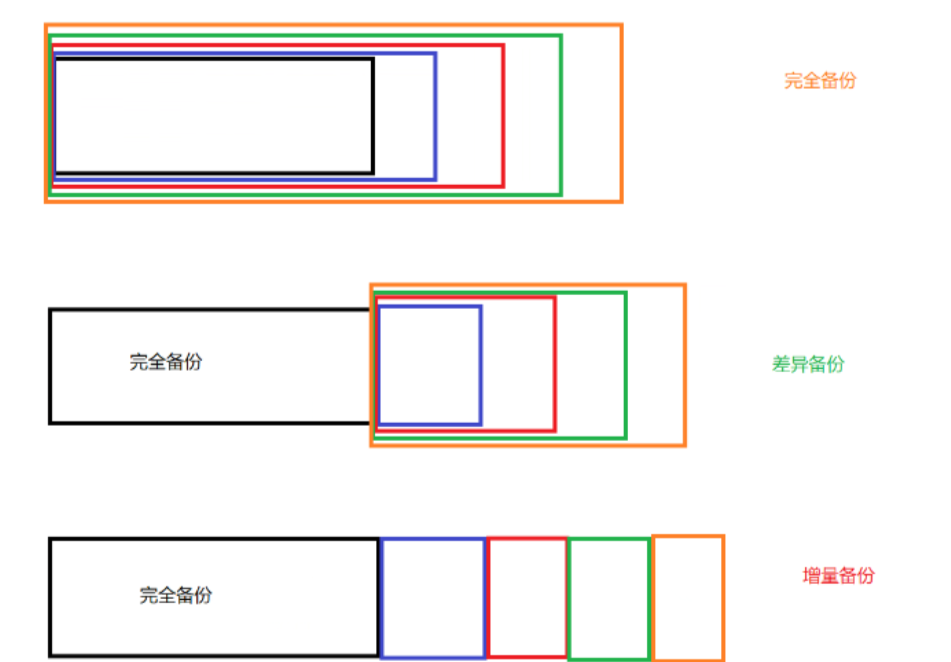
温备份: 数据库锁定表格 (不可写入但可读)的状态下进行备份操作

逻辑备份:对数据库逻辑组件 (如: 表等数据库对象)的  
备份

**2、备份策略角度**

**从数据库的备份策略角度，备份可分为**

* 完全备份:每次对数据库进行完整的备份
* 差异备份: 备份自从上次完全备份之后被修改过的文件
* 增量备份:只有在上次完全备份或者增量备份后被修改的文件才会被备份



## 二、常见的备份方法

**物理冷备**

备份时数据库处于关闭状态，直接打包[数据库文件](https://so.csdn.net/so/search?q=%E6%95%B0%E6%8D%AE%E5%BA%93%E6%96%87%E4%BB%B6&spm=1001.2101.3001.7020)  
备份速度快，恢复时也是最简单的

**专用备份工具mysqldump或mysqlhotcopy**

mysqldump常用的逻辑备份工具  
mysqlhotcopy仅拥有备份MyISAM和ARCHIVE表

**启用二进制日志进行增量备份**

进行增量备份，需要刷新二进制日志

备份工具

tar压缩打包（完全备份，物理冷备）

mysqldump（完全备份，逻辑热备）

mysqlhotcopy（完全备份，逻辑热备，仅支持MyISAM和ARCHIVE引擎表）

二进制日志（增量备份）

PXB XtraBackup innobackupex（完全备份、增量备份，物理热备

### MySQL物理冷备份及恢复



### mysqldump备份数据库

### 

### mysqldump备份数据表

### 、

## 三、MySQL完全备份

### 1、 什么是完全备份

完全备份是对整个数据库、数据库结构和文件结构的备份  
保存的是备份完成时刻的数据库  
是差异备份与增量备份的基础

### 2、完全备份的优缺点

优点：

备份与恢复操作简单方便  
缺点：

数据存在大量的重复  
占用大量的备份空间  
备份与恢复时间长

### 3、完全备份的方法

### （1）物理冷备份与恢复

### 关闭MySQL数据库

### 使用tar命令直接打包数据库文件夹

### 直接替换现有MySQL目录即可

### （2）mysqldump备份与恢复

### MySQL自带的备份工具，可方便实现对MySQL的备份

### 可以将指定的库、表导出为SQL脚本

### 使用命令mysq|导入备份的数据

#### 4、MySQL物理冷备份及恢复操作  第一步：物理冷备份

### [root@localhost ~]# systemctl stop mysqld #先关闭服务

### 

### [root@localhost ~]# cd /usr/local/mysql/

### 

### [root@localhost mysql]# tar zcf /opt/mysql\_backup\_$(date +%Y%m%d) data/

### #将数据库的data/文件进行压缩备份

### 

#### 第二步：恢复数据库（或迁移数据库）

[root@localhost mysql]# cd /opt/

[root@localhost opt]# ls

mysql-5.7.44 mysql\_backup\_20240620 mysql-boost-5.7.44.tar.gz rh

[root@localhost opt]# scp mysql\_backup\_20240620 192.168.10.20:/opt

#将压缩好的备份文件发送到目标主机

#在目标主机中将旧数据库文件替换为新数据库文件

[root@localhost ~]# cd /opt

[root@localhost opt]# ls

mysql-5.7.44 mysql\_backup\_20240620 mysql-boost-5.7.44.tar.gz rh

[root@localhost opt]# systemctl stop mysqld

[root@localhost opt]# tar xf mysql\_backup\_20240620 #解压缩发送来的备份文件

[root@localhost opt]# ls

data mysql-5.7.44 mysql\_backup\_20240620 mysql-boost-5.7.44.tar.gz rh

[root@localhost opt]# cd /usr/local/mysql/

[root@localhost mysql]# ls

bin data docs include lib LICENSE man mysql-test README README-test share support-files usr

[root@localhost mysql]# mv data/ data.bak

[root@localhost mysql]# ls

bin data.bak docs include lib LICENSE man mysql-test README README-test share support-files usr

[root@localhost mysql]# mv /opt/data/ /usr/local/mysql/ #data/替换为发送端发送来的数据库文件

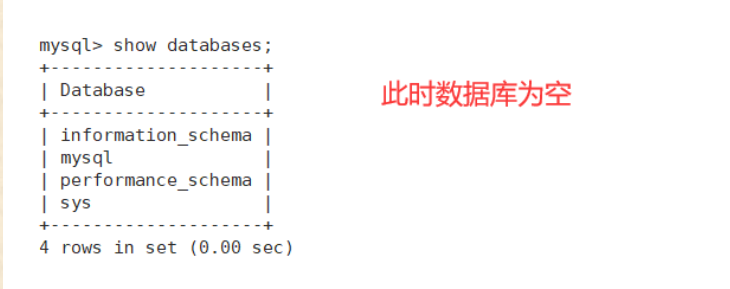
[root@localhost mysql]# ls

bin data data.bak docs include lib LICENSE man mysql-test README README-test share support-files usr

[root@localhost mysql]# ls data

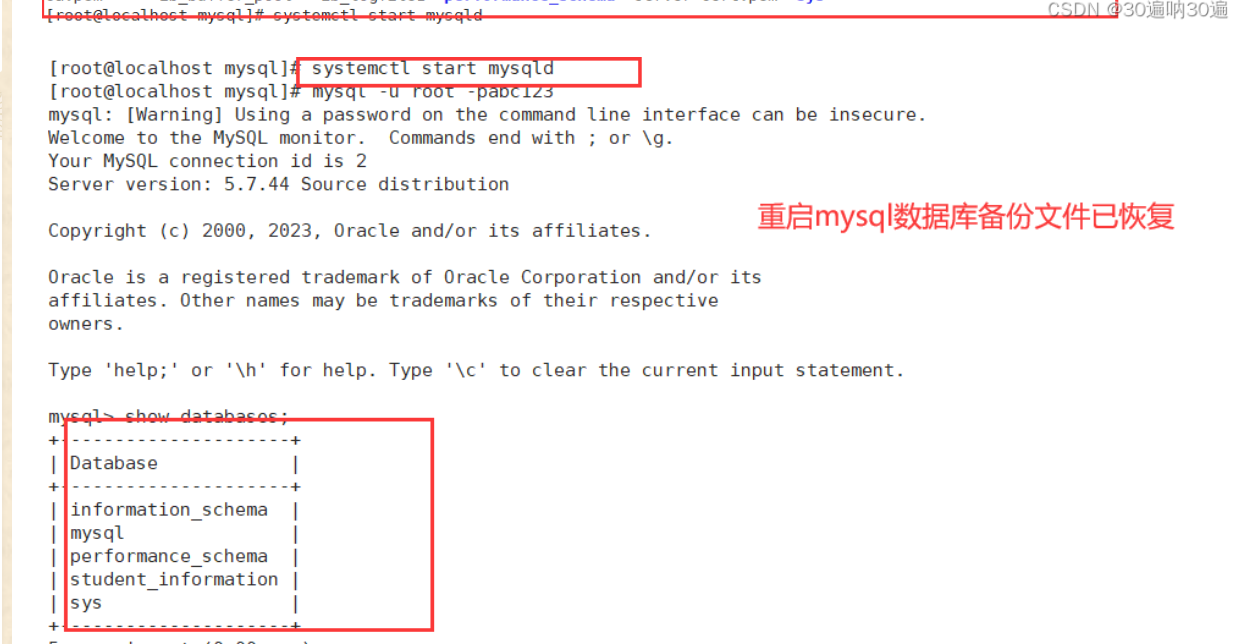
auto.cnf client-cert.pem ibdata1 ibtmp1 private\_key.pem server-key.pem

ca-key.pem client-key.pem ib\_logfile0 mysql public\_key.pem student\_information

ca.pem ib\_buffer\_pool ib\_logfile1 performance\_schema server-cert.pem sys

[root@localhost mysql]# systemctl start mysqld # 重启数据库





### 5、mysqldump备份数据库

#### ****（1）mysqldump命令对单个库进行完全备份****

**mysqldump -u 用户名-p[密码][选项][数据库名] >/备份路径/备份文件名**

### mysqldump -u root -pabc123 student\_information > /opt/backup/student\_information.sql

### mysqldump -u root -pabc123 --databases student\_information >/opt/backup/student\_information1.sql

### **注：加上--databases备份文件中会包含库的操作对象的命令，而不加--databases则不会包含**。

### [root@localhost ~]# cd /opt/

### [root@localhost opt]# mkdir backup

### [root@localhost opt]# cd backup/

### [root@localhost backup]# pwd

### /opt/backup

### [root@localhost backup]# ls

### 

### [root@localhost backup]# mysqldump -u root -pabc123 student\_information > /opt/backup/student\_information.sql

### 

### mysqldump: [Warning] Using a password on the command line interface can be insecure.

### [root@localhost backup]# ls

### student\_information.sql

### 

### [root@localhost backup]# mysqldump -u root -pabc123 --databases student\_information > /opt/backup/student\_information1.sql

### 

### mysqldump: [Warning] Using a password on the command line interface can be insecure.

### [root@localhost backup]# ls

### student\_information1.sql student\_information.sql

#### （2）mysqldump命令对多个库进行完全备份

### mysqldump -u 用户名 -p[密码][选项] --databases 库名 1:[库名2]I... > /备份路径/备份文件名

#### (3)mysqldump命令对所有库进行完全备份

mysqldump -u 用户名-p[密码]选项]--all-databases > /备份路径/备份文件名

### [root@localhost backup]# mysqldump -u root -pabc123 --all-databases >/opt/backup/all\_databases.sql

### mysqldump: [Warning] Using a password on the command line interface can be insecure.

### [root@localhost backup]# ls

### all\_databases.sql student\_information1.sql student\_information.sql

### [root@localhost backup]# cat all\_databases.sql | grep "^USE"

### USE `mysql`;

### USE `student\_information`;

### 6、mysqldump备份数据表

### **对库内特定的表进行备份**

### **mysqldump -u 用户名-p[密码][选项] 数据库名 表名>/备份路径/备份文件名**

### **[root@localhost backup]# mysqldump -u root -pabc123 student\_information a1 a2 > /opt/backup/a12.sql**

### **mysqldump: [Warning] Using a password on the command line interface can be insecure.**

### **[root@localhost backup]# vim /opt/backup/a12.sql**

### **#备份文件中只有关于a1和a2表的操作**

### 

### 

### 

## 四、数据库数据恢复

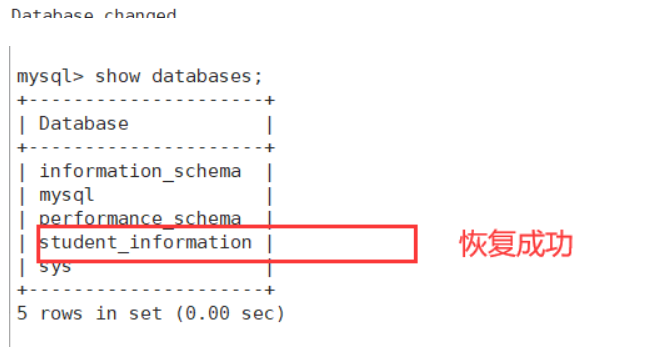
### 1、库恢复

**使用mysqldump导出的脚本，可使用导入的方法**

source命令  
mysql命令





、

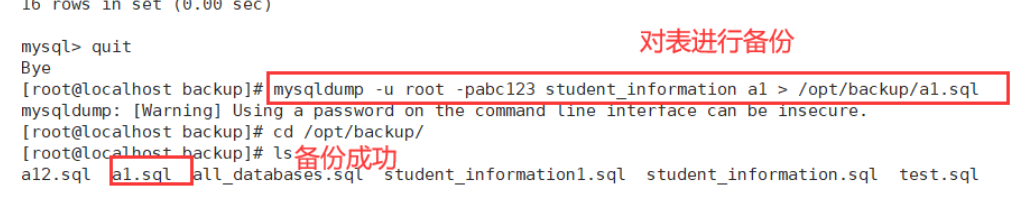
**（2）使用mysql命令恢复数据**

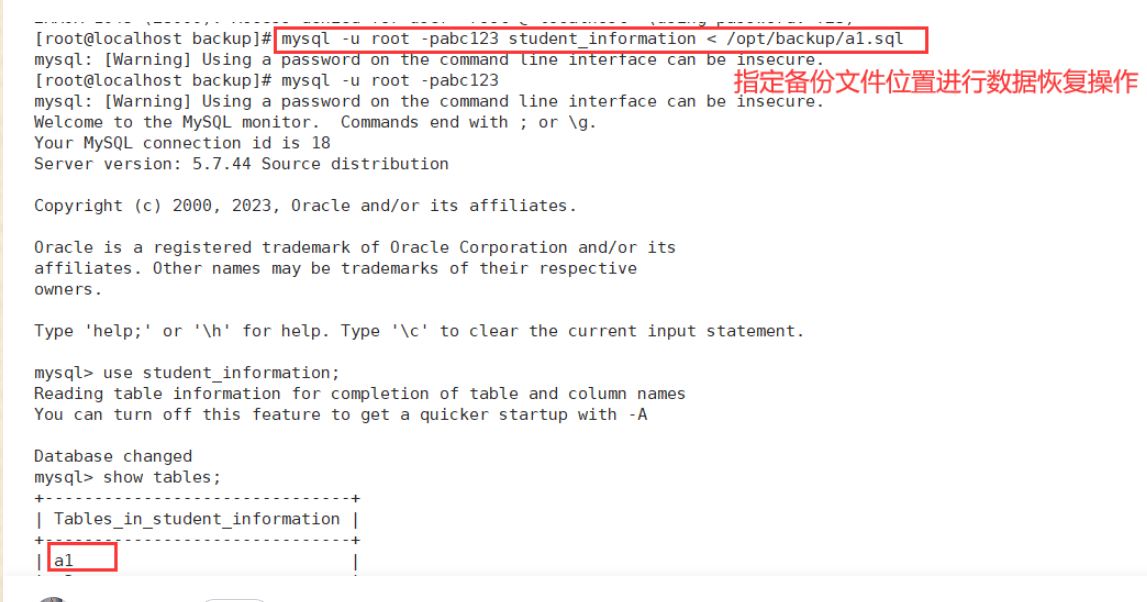


### 2、表恢复

恢复表时同样可以使用source或者mysql命令  
source恢复表的操作与恢复库的操作相同  
当备份文件中只包含表的备份，而不包括创建库的语句时，必须指定库名，且

目标库必须存在





## 完全备份和恢复小结

**完全备份**

物理冷备：先关闭数据库，使用 tar 命令压缩打包备份数据库的数据目录和文件 mysql/data/

mysqldump 逻辑热备

mysqldump -u root -p密码 --databases 库1 [库2 ....] > XXX.sql #备份一个或多个指定的库及库中所有的表

mysqldump -u root -p密码 --all-databases > XXX.sql #备份所有库

mysqldump -u root -p密码 库名 > XXX.sql #只备份指定库中的所有表（不包含库对象本身）

mysqldump -u root -p密码 库名 表1 [表2 ....] > XXX.sql #只备份指定库中的一个或多个指定的表（不包含库对象本身

**完全恢复**

1）先登录数据库，再执行 source sql文件路径 （如何sql文件里只备份了表，需要先 use 切换库再执行 source）

2）mysql -u root -p密码 < sql文件路径 cat sql文件路径 | mysql -u root -p密码 #恢复库

mysql -u root -p密码 库名 < sql文件路径 cat sql文件路径 | mysql -u root -p密码 库名 #恢复表

————————————————

五、增量备份与恢复

使用mysqldump进行完全备份存在的问题

备份数据中有重复数据

备份时间与恢复时间过长

增量备份的特点

没有重复数据，备份量不大，时间短

恢复需要上次完全备份及完全备份之后所有的增量备份才 能恢复，而且要对所有增量备份进行逐个反推恢复

（2）增量备份的过程

1、MySQL没有提供直接的增量备份方法

2、可通过MySQL提供的二进制日志间接实现增量备份

3、MySQL二进制日志对备份的意义

二进制日志保存了所有更新或者可能更新数据库的操作

二进制日志在启动MySQL服务器后开始记录，并在文件达到 max\_binlog\_size所设置的大小 或 者接收到flush logs命令后重新 创建新的日志文件

只需定时执行flush logs方法重新创建新的日志，生成二进制文件序列，并及时把这些日志保存到安全的地方就完成了一个时间段的增量备份

————————————————

### MySQL日志管理介绍

----------------------MySQL 日志管理--------------------------------

MySQL 的日志默认保存位置为 /usr/local/mysql/data

vim /etc/my.cnf

[mysqld]

##错误日志，用来记录当MySQL启动、停止或运行时发生的错误信息，默认已开启

log-error=/usr/local/mysql/data/mysql\_error.log #指定日志的保存位置和文件名

##通用查询日志，用来记录MySQL的所有连接和语句，默认是关闭的

general\_log=ON

general\_log\_file=/usr/local/mysql/data/mysql\_general.log

##二进制日志(binlog)，用来记录所有更新了数据或者已经潜在更新了数据的语句，记录了数据的更改，可用于数据恢复，默认已开启

log-bin=mysql-bin #也可以 log\_bin=mysql-bin

##慢查询日志，用来记录所有执行时间超过long\_query\_time秒的语句，可以找到哪些查询语句执行时间长，以便于优化，默认是关闭的

slow\_query\_log=ON

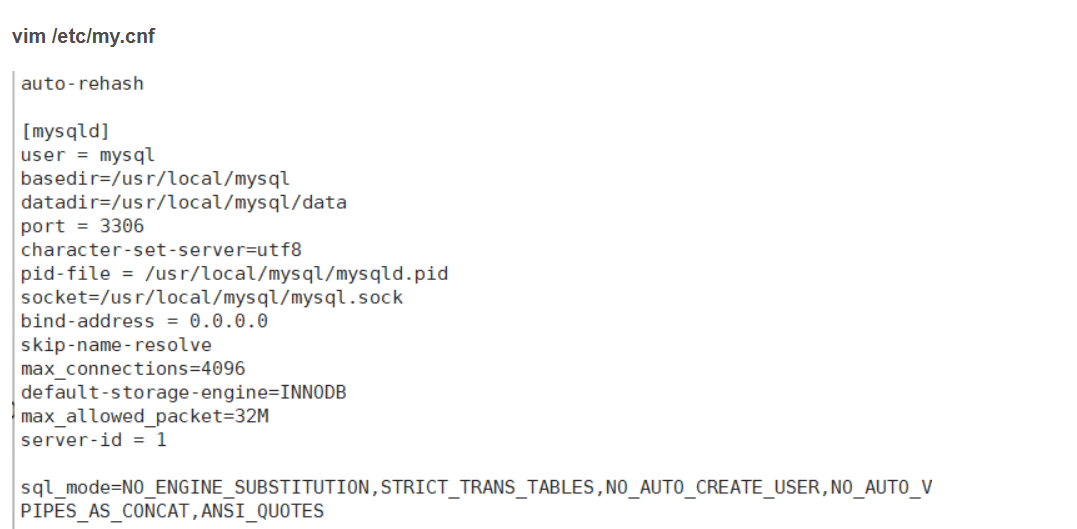
slow\_query\_log\_file=/usr/local/mysql/data/mysql\_slow\_query.log

long\_query\_time=5 #设置超过5秒执行的语句被记录，缺省时为10秒

————————————————

### 增量备份操作

**在进行增量备份前需要修改主配置文件：**







二进制日志(binlog)有3种不同的记录格式:

STATEMENT (基于SQL语句)、ROW(基于行)、MIXED(混合模式)，默认格式STATEMENT ​

STATEMENT (基于SQL语句)：记录修改的sql语句。高并发的情况下，记录操作的sql语句时可能 导致顺序会有错误，导致恢复数据时，数据丢失或有误差。效率高，但数据可能有误差。

ROW(基于行)：记录每一行数据，准确，但恢复时效率低。

​MIXED(混合模式)：正常情况下使用STATEMENT，高并发的情况下会智能地切换到ROW。

#### 、具体实现演示

步骤一：

创建两个空文件用于存放完全备份文件和增量备份文件

### 

### 

### 

### 

### 

### 

### 

### 